

## 優先権付き待ち行列網の近似解法

紀 一 誠 (Issei KINO)

日本電気(株) C&Cシステム研究所 応用システム研究部

### 1. はじめに

優先権制御を行なうノードを含む待ち行列網の解析は、計算機システムの性能評価モデルとして重要な役割をもつため、モデル利用者からの期待と要請は根強いものがある。しかし、この種の待ち行列網は積形式解を持たない事が良く知られており、解法は近似解法に頼らざるを得ない。積形式解をもたない網に関する近似解法は、Chandy et al.<sup>1)</sup>によるマトリックス法あるいは等価流量法(flow-equivalent method)とよばれる方法、Marie<sup>2)</sup>の方法、これらと組み合わせた方法<sup>3)</sup>等が知られている。この内、等価流量法は基礎となる近似法として位置付けられている。等価流量法は、積形式タイプのノード群をひとまとめにし、そのスループットと等価な状態依存形のサービス率をもつ1個のノード $Q_c$  (composite node) に置換し、 $Q_c$ と非積形式ノードから構成される網を数値的に解く方法である。

等価流量法は、たたみこみ演算と基本算法とする計算法に

なじみ易いという利点をもっている。しかし、等価流量法を用いた場合、積形式タイプのノードに関する諸量を得るためには、 $Q_0$ と非積形式ノードから成る網を解いて得られる状態確率を利用して積形式ノードのみから成る網に関するたたみこみ結果を重味付けしなればならず、網の規模が大きくなるに従ってこの部分の計算量は飛躍的に増大してしまうという難点に出会う。

本稿では、等価流量法を用いる際に発生するこの難点を避けるための計算法について述べる。本稿の方法は、たたみこみ結果を状態確率で重味付けする従来の方法に代えて、優先権制御を行なうノード(非積形式ノード)におけるスループットと等しいスループットを実現するような仮想的な積形式ノード(SRadaw Node)を構成し、このSN(SRadaw Node)を非積形式ノードに代えて網に接続し、全体を積形式解をもつ待ち行列網として解く方法である。SNを構成する段階で非線型の連立方程式を解く必要に迫られるが、解法に用いるNewton-Raphson法の2次収束性のため、演算量は従来の方法に比べて大中に削減できる。

## 2. モデル

$N+1$ 個のノードから成る待ち行列網 $Q$ を考える。ノード $i$ を $Q_i$ とし、 $Q_0$ を優先権制御を行なうノード、 $Q_1, \dots, Q_N$ を積形式ノード

とする。網内には  $L$  個の閉鎖型部分連鎖に従い移動をえる  $L$  種類の客が存在するものとし、その客数ベクトルを  $\mathbf{K} = (k_1, \dots, k_L)$  とする。部分連鎖  $j$  に従う客をクラス  $j$  の客とよび、 $Q_0$  におけるクラス  $j$  の客のサービス要求時間はパラメータ  $\mu_j$  なる指数分布に従うものとする。 $Q_0$  はサーバ数  $m$  の複数窓口とし、優先順位は若番クラスの客ほど高いものとする。制御方式は、割込優先・中断点再開方式 (preemptive-resume) とし、同一順位内は先着順とする。割込・中断は常に最下位順位の客に対して行われるものとする。図 1. にモデルを示す。

クラス  $j$  の客のノード  $i (1, \dots, N)$  への相対訪問回数  $\theta_{ij}$  とし、ノード 0 への相対訪問回数  $\theta_{0j} = 1, j = 1, 2, \dots, L$  とする。また、 $r_j = 1/\mu_j, \rho_{0j} = \theta_{0j}/\mu_j = r_j$  とし、 $Q_0$  の負荷ベクトル  $\rho_0 = (\rho_{01}, \rho_{02}, \dots, \rho_{0L}) = (r_1, r_2, \dots, r_L)$  とする。さらに、 $\alpha, \alpha$   $L$  次元ベクトルとし、次の記法を定義しておく。

$$\alpha = \alpha_1 + \alpha_2 + \dots + \alpha_L, \quad \alpha = \alpha_1^{\alpha_1} \alpha_2^{\alpha_2} \dots \alpha_L^{\alpha_L},$$

$$\alpha! = \alpha_1! \alpha_2! \dots \alpha_L!.$$

次に、待ち行列網  $Q$  から  $Q_i$  を取り除いてできる待ち行列網を  $Q$  に対して、 $i$ -complement 網とよび、 $Q_{[i]}$  と記すことにする。本稿のモデルでは、0-complement 網  $Q_{[0]}$  は積形式解をもつ待ち行列網になる。 $Q_{[0]}$  に関する正規化定数は  $G_{[0]}(\mathbf{R}), 0 \leq \mathbf{R} \leq \mathbf{K}$  と記すことにする。

3.  $Q_0$  の状態確率

等価流量法を用いて  $Q_0$  の状態確率を求める方法について示す。

図2. に示すように、等価流量法ではまず、 $Q_0$  に代えてサービス時間  $0$  をも

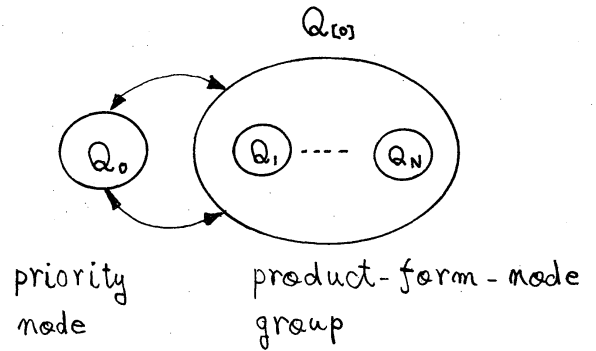


Figure 1. Priority Queuing Network

つ仮想的な積形式ノード  $Q_{101}$  に接続することにより、 $Q_0$  におけるクラス  $j$  の密のフルジョットの近似値  $\tau_j(\mathbf{r})$  を次の如く求める。但し、 $\mathbf{e}_j$  は  $j$  方向の単位ベクトル、 $j=1, 2, \dots, L$  とする。

$$(3.1) \quad \tau_j(\mathbf{r}) = G_{101}(\mathbf{r} - \mathbf{e}_j) / G_{101}(\mathbf{r}), \quad 0 \leq \mathbf{r} \leq \mathbf{K}$$

この過程は  $Q_0$  をあたかもショートしたかのように見えるため、電気回路理論の Norton (又は Thévenin) の定理になぞらえられている。

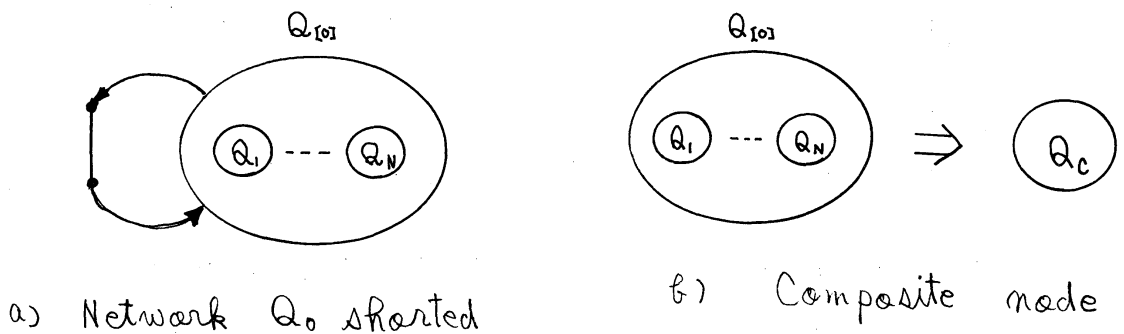


Figure 2. Reduction of the original network to the equivalent one.

次に,  $Q_{i0}$  とスループット等価な

ノード  $Q_c$  を構成する.

$Q_c$  は, その状態が  $k$  の時に  
クラス  $j$  の客に対するサービス

率が (3.1) に示される  $\tau_j(k)$

となる状態依存形のサービス

率をもつノードとして構成さ

れる. こうして, 元の網  $Q$  は図3. に示される如くに,  $Q_0$  と

$Q_c$  の2つのノードから成る網  $(Q_0, Q_c)$  に縮約する事ができる.

$Q_0$  の状態が  $k$  の時, クラス  $j$  の客に関するサービス率を  $\mu_j(k)$  と  
すれば本稿のモデルでは次の如くに定義される.

$$(3.2) \quad \mu_j(k) = \begin{cases} \mu_j \times \min(m, r_1) & , j=1 \\ \mu_j \times \min([m - (r_1 + r_2 + \dots + r_{j-1})]^+, r_j) & , j=2, \dots, L. \end{cases}$$

但し,  $[x]^+ = (x, 0 \leq x; 0, x < 0)$  とする.

待ち行列網  $(Q_0, Q_c)$  は積形式解をもつ待ち行列網とはなら

ないが, 高々2個のノードしか存在しないので, 状態  $k, 0 \leq k \leq K$

に関する状態方程式を作りこれを数値的に解くことにより,

$Q_0$  に関する状態確率を計算することが出来る. この部分の

計算は本稿のモデルの場合, 複合構造をもつブロック3重対角

行列を係数とする連立方程式となり, この特殊な構造を利用して

未知数の数を削減し効率的な計算法を作る事ができる.

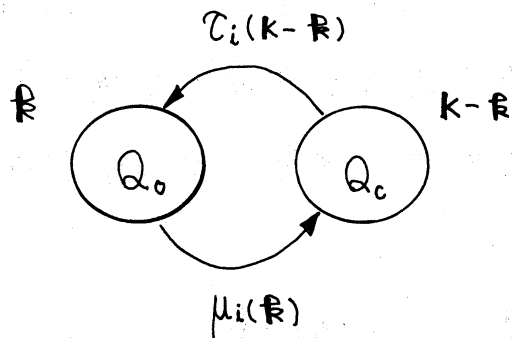


Figure 3. Reduced network

各クラスの客について， $Q_{01}$ に入ってから出て来るまでの細内滞在時間の分布を指数分布と仮定して $(Q_0, Q_c)$ を解いている点が等価流量法が近似解法である理由である。

#### 4. 等価流量法による $Q_{01}$ システムの計算法

$Q_0$ に関する状態確率を $p_0(k)$ とする。 $p_0(k)$ が得られた後， $Q_i (i=1, 2, \dots, N)$ におけるクラス $j$ の客の待ち行列長(含サービス中) $L_{ij}(k)$ の計算を考える。 $Q_c$ の状態が $k$ の時 $Q_0$ の状態は $k-k$ に注意し， $M_{ij}(k)$ を $Q_{01}$ においてその細内容数ベクトルが $k$ である時のロード $i$ でのクラス $j$ の客の待ち行列長とすれば， $L_{ij}(k)$ は， $M_{ij}(k)$ と状態確率で重味付けし次式により得られる。

$$(4.1) \quad L_{ij}(k) = \sum_{0 \leq k \leq k} M_{ij}(k) p_0(k-k)$$

計算手順をアルゴリズムFとし，以下にプログラム風に展開する。

```

F
  For i = 1 to N
    F1 :  $G_{10, i1}(k)$ ,  $0 \leq k \leq K$  作成
    For j = 1 to L
      For k = 0 to K
        F3 :  $M(k) \leftarrow \sum_{0 \leq x \leq k} x_j g_i(x) G_{10, i1}(k-x) / G_{101}(k)$ 
      Next k
    F4 :  $L_{ij}(K) \leftarrow \sum_{0 \leq x \leq K} M(k) p_0(K-k)$ 
  Next j
Next i
  
```

但し,  $q_i(x)$  は  $Q_i$  の状態が  $x$  である正規化されない状態確率とする. アルゴリズム F の演算量と大雑把に見積る. 本例では(乗算回数)  $\doteq$  (加算回数)であるので, 乗算回数を演算量の目安にすることにする. 乗算回数は次に与えられる.

$$(4.2) \quad (\text{乗算回数})_F = NR \left[ 1 + L \left\{ \prod_{j=1}^L (k_j + 2) / 2^L + 1 \right\} \right]$$

但し,  $R$  は  $0 \leq R \leq K$  なる定義域上で行われるたたみこみ演算に必要とされる乗算回数で,  $R \doteq \prod_{j=1}^L (k_j + 1)$  である.

## 5. Shadow Node 接続法による $Q_{i0}$ システムの計算法

### 5.1 Shadow Node $Q_s$ の構成法

$Q_{i0}$  に関する情報を得るために,  $Q_0$  の動きを模倣する積形式ノード  $Q_s$  を構成し,  $Q_0$  に代えて  $Q_s$  を  $Q_{i0}$  に接続し  $(Q_s, Q_{i0})$  を全体として積形式解をもつ待ち行列網として解く事により,  $Q_i, i=1, \dots, N$  に関する情報を得る.

$Q_s$  の構成は以下の如くである.  $Q_s$  は図4に示すようなフィードバック・ループをもつタイプ1のノードとする. 即ち,  $Q_s$  におけるサービス要求時間は各クラスとも共通のパラメータ  $\mu_s$  をもつ指数分布に従うものとし, サービス規律は全クラスを通じての先着順とする. また,  $Q_s$  に  $R$  人の客が存在する時のサービス率を  $\mu_s(R)$  とする.  $\gamma_j$  をクラス  $j$  の客がサービス終了後  $Q_{i0}$  に進む確率とすれば, クラス  $j$  の客の  $Q_s$  における平均ループ回数  $\theta_{sj}$  は

$$\theta_{sj} = 1 / (1 - r_j) \quad \text{となる.}$$

また、クラス  $j$  の客が  $Q_s$  に到着してから  $Q_s$  を離れるまでの間に受ける延べサービス時間の平均値を  $\rho_{sj}$  とすれば次の関係がある。

$$(5.1) \quad \rho_{sj} = \theta_{sj} / \mu_s.$$

$Q_0$  の動きを模倣させるために変化させるパラメータは

$r_j, j=1, \dots, L$  であるが、これらは (5.1) により  $\rho_{sj}$  に変換できるので以降は直接  $\rho_{sj}$  を変化させるものと考えることとする。

$Q_s$  は  $Q_0$  とスループット等価となる事を規範として構成されるものとする。即ち、 $\rho_{sj}$  を適当に選ぶ事により、各クラス毎のスループットを  $Q_0$  のそれに一致するように  $Q_s$  は構成されるものとする。しかし、スループット等価規範だけでは積形式ノード  $Q_s$  を一意に定める事はできず、さらに最適性の条件が必要となる。

この点に関しては後に述べる。

### 5.2 スループットに関する方程式

$(Q_0, Q_0)$  を解いた結果得られる  $Q_0$  におけるクラス  $j$  の客のスループット  $\varphi_j$  は次式で与えられる。

$$(5.2) \quad \varphi_j = \sum_{0 \leq k \leq K} \mu_j(k) p_0(k).$$

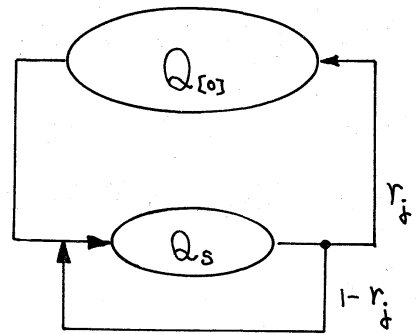


Figure 4. FIFO made with feedback loops.



スル-ポット・ベクトル  $\mathcal{G}$ ,  $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_L)$  とする. 一方,  
 $Q_s$  の状態が  $k$  である正規化された状態確率  $g_s(k)$  は次式の  
 如くに表現できる.

$$(5.3) \quad g_s(k) = Q_s(|k|) \frac{|k|!}{k!} \rho_s^k$$

但し,  $\rho_s = (\rho_{s1}, \rho_{s2}, \dots, \rho_{sL})$  とし,  $Q_s(k)$  を次に定義される  $Q_s$   
 の容量係数とする.

$$(5.4) \quad Q_s(k) = \left\{ \prod_{j=1}^L \mu_s(j) \right\}^{-1}, \quad Q_s(0) = 1.$$

待ち行列網  $(Q_s, Q_{T0j})$  は積形式解をもつ網として解く事が  
 できる. この網に関する正規化定数は次式に与えられる.

$$(5.5) \quad G(k) = \sum_{0 \leq x \leq k} g_s(x) G_{T0j}(k-x), \quad 0 \leq k \leq K.$$

$Q_s$  におけるクラス  $j$  の容のスル-ポット  $\nu_j$  はこの正規化定数を用  
 いて次の如くに得られる.

$$(5.6) \quad \nu_j = \theta_{sj} G(k - e_j) / G(k).$$

スル-ポット・ベクトル  $\mathcal{V}$ ,  $\mathcal{V} = (\nu_1, \nu_2, \dots, \nu_L)$  とする.

$Q_0$  にスル-ポット等価な  $Q_s$  を構成するといふ事は,  $Q_0$  のスル-ポット  $\mathcal{G}$   
 が与えられた時, 次に示されるスル-ポット方程式を解き, 解ベ  
 クトル  $\rho_s$  を求めることに他ならない.

$$(5.7) \quad \mathcal{V} = \mathcal{G}.$$

上式は (5.3) ~ (5.6) から明らかなるように,  $\rho_s$  に関する  $L$  元の  
 非線型連立方程式となっている.

5.3 スル-ポット方程式の解法

方程式(5.7)は基本的には Newton-Raphson法を用いて解く事ができるが、解の存在領域として、 $0 \leq \rho_s$  なる制約条件が付いているため、この制約条件を取り除くための若干の工夫が必要とされる。以下簡略化のため  $\rho_s$  の添字  $s$  を省略する。

繰り返しの  $n$  ステップ時の解を  $\rho^{(n)}$  とすれば、Newton-Raphson法の計算スキームは次の如くに表現される。

$$(5.8) \quad \rho^{(n+1)} = \rho^{(n)} - \{ \Psi(\rho^{(n)}) - \Psi \} \left[ \frac{\partial \Psi}{\partial \rho} \right]_{\rho = \rho^{(n)}}^{-1}$$

但し、 $[\partial \Psi / \partial \rho]$  は次に定義されるヤコビ行列マトリクスとする。

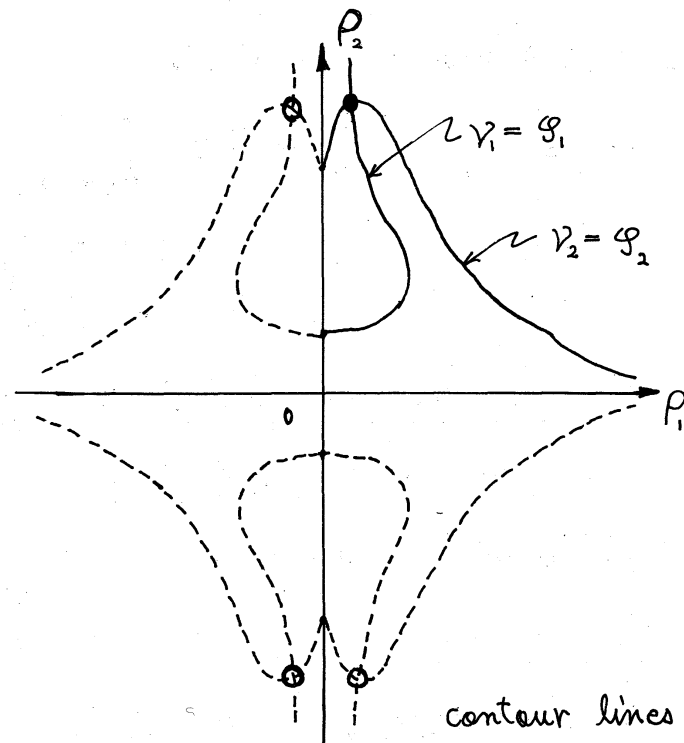
$$(5.9) \quad \left[ \frac{\partial \Psi}{\partial \rho} \right] = \begin{bmatrix} \frac{\partial \Psi_1}{\partial \rho_1} & \frac{\partial \Psi_2}{\partial \rho_1} & \cdots & \frac{\partial \Psi_L}{\partial \rho_1} \\ \frac{\partial \Psi_1}{\partial \rho_L} & \frac{\partial \Psi_2}{\partial \rho_L} & \cdots & \frac{\partial \Psi_L}{\partial \rho_L} \end{bmatrix}$$

制約条件  $0 \leq \rho$  を外すために  $\Psi(\rho)$  を、 $-\infty < \rho < +\infty$  で定義された関数  $\hat{\Psi}(\rho)$  に次の如く拡張する。

$$(5.10) \quad \hat{\Psi}(\rho_1, \rho_2, \dots, \rho_L) = \Psi(|\rho_1|, |\rho_2|, \dots, |\rho_L|)$$

この拡張された関数  $\hat{\Psi}$  により、(5.8)に示される計算スキームを適用して計算を行なう。もし、(5.7)の解が  $0 \leq \rho$  なる範囲に一意に存在し、かつ  $\hat{\Psi}$  に関する(5.8)が収束するならば、(5.10)より明らかなる関数の対称性から、ファントム解の符号のみを正とする事によって容易に真の解を得る事ができる。

図5. に  $L=2$  の場合のファントム解と真の解との関係を示す例を示す。図5. では、 $\rho_1, \rho_2$  平面上に、 $\nu_1 = \varphi_1$ 、 $\nu_2 = \varphi_2$  となるような等高線と、 $\xi$  による (5.10) により全平面に拡張したものが示される。



- : real solution
- : phantom solutions

交点のうち、正のもののみが真の解となり、その他はファントム解となる。

Figure 5. Real solution and phantom solutions.

計算スキーム (5.8) を実行するためには  $\partial G(k) / \partial \rho_j$  を導く必要がある。以下にこれを導く。

$$(5.11) \quad \frac{\partial \nu_i}{\partial \rho_j} = \frac{\partial s_i}{G(k)^2} \{ G(k) \frac{\partial}{\partial \rho_j} G(k - e_i) - G(k - e_i) \frac{\partial}{\partial \rho_j} G(k) \}.$$

また、 $\partial G(k) / \partial \rho_j$  は次の如くとなる。

$$(5.12) \quad \begin{aligned} \frac{\partial}{\partial \rho_j} G(k) &= \sum_{0 \leq r \leq k} \frac{\partial}{\partial \rho_j} a_s(|R|) \frac{|R|!}{r!} \rho_j^R G_{[r]}(k-r) \\ &= \sum_{0 \leq r \leq k} R_j a_s(r) G_{[r]}(k-r) / \rho_j. \end{aligned}$$

以上をまとめて次が得られる。

$$(5.13) \quad \frac{\partial \nu_i}{\partial \rho_j} = \left\{ \theta_{si} \sum_{0 \leq k \leq K} R_j \varphi_s(k) G_{[0]}(K - e_i - k) - \nu_i \sum_{0 \leq k \leq K} R_j \varphi_s(k) G_{[0]}(K - k) \right\} / \rho_j G(k).$$

#### 5.4 SShadow Node の構成アルゴリズム

前節の解法をアルゴリズム S として以下にまとめる。但し、 $\Delta \rho^{(n+1)} = \rho^{(n+1)} - \rho^{(n)}$  とする。

##### アルゴリズム S

S1: 初期設定,  $\rho \leftarrow \rho^{(0)}$

S2: たたみこみ,  $G(k), 0 \leq k \leq K$  作成

S3: スル-ジョット計算,  $\nu_j \leftarrow \theta_{sj} G(K - e_j) / G(K), j = 1, 2, \dots, L.$

S4: ヤコビアン・マトリクスの作成,  $i, j = 1, 2, \dots, L,$

$$[\partial \nu / \partial \rho] \leftarrow \{ \partial \nu_i / \partial \rho_j \}.$$

S5: 連立方程式を解き,  $\rho^{(n+1)}$  を求める。

$$\nu - \varphi = -\Delta \rho^{(n+1)} [\partial \nu / \partial \rho].$$

S6: 収束判定,  $\max_j |(\nu_j - \varphi_j) / \varphi_j| < \varepsilon.$

満足したら終了, しない場合には S2 にもどる。

#### 5.5 最適 SShadow Node

先に述べた如く, スル-ジョット  $\varphi$  を実現する積形式ノード  $Q_s$  は一意に定まるわけではなく, 異なるノードタイプを用いても同一の  $\varphi$  を実現できる事ができる。5.1 節に示すように,

$Q_s$  としてタイプ1のノードを選んだとしてもその状態依存形のサービス率の形が異なるものを用いてやはり同一の $\Phi$ を実現する事ができる。従って、 $Q_s$ を一意に定めるためには何らかの最適条件を加えなければならぬ。本稿ではこの最適条件として、 $Q_0$ における滞在時間に最も近い滞在時間を実現する $Q_s$ を最適とする、という条件を採用することにする。

即ち、 $(Q_0, Q_c)$ を解いて得られるノード0のクラス $j$ の客の滞在時間を $\hat{S}_j$ とし、スループット等価な $Q_s$ と $Q_{\text{reg}}$ からなる網 $(Q_s, Q_{\text{reg}})$ を解いて得られるクラス $j$ の客の $Q_s$ における滞在時間を $S_j$ とした時、 $\|S - \hat{S}\|$ を最小とするような $Q_s$ を最適なノードとする。但し、 $S = (S_1, \dots, S_L)$ 、 $\hat{S} = (\hat{S}_1, \dots, \hat{S}_L)$ とし、 $\|x\| = |x|$ とする。最適 Shadow Nodeを $Q_s^*$ と表わすこととする。

Conjecture:  $Q_s$ のサービス率を示す関数を次とする。

$$(5.14) \quad \mu_s(r) = \begin{cases} r \mu_s & , \quad r < M \\ M \mu_s & , \quad M \leq r \end{cases}$$

この時、 $Q_s^*$ は $M = m$ なる $\mu_s(r)$ をもつノードである。

上記の予想は後に示す数値例により確認する事ができる。

## 6. 解法アルゴリズム

前章までに示した解法について、全体を通しての計算手続を以下にまとめておく。

Step 1 : 初期設定

$G_{IOJ}(k)$ ,  $0 \leq k \leq K$  を作成.

Step 2 :  $Q_c$  の構成 (係数行列  $A$  の作成)

$\mu_j(k)$ ,  $c_j(k-k)$ ,  $0 \leq k \leq K$ , から  $A$  を作成.

Step 3 :  $(Q_0, Q_c)$  システム の計算

$x_0 A = 0$ ,  $|x_0| = 1$  を解く.

Step 4 :  $Q_0$  に関する諸量の計算

スループット  $\psi$ , 滞在時間  $\hat{S}$ , 他.

Step 5 :  $Q_s^*$  の構成

$M = m$  とし,  $\psi = \psi$  を アルゴリズム S を用いて解く.

Step 6 :  $(Q_s^*, Q_{IOJ})$  システム の計算.

正規化定数  $G(k)$ ,  $0 \leq k \leq K$  の計算.

Step 7 :  $Q_{IOJ}$  に関する諸量の計算

$Q_1, Q_2, \dots, Q_N$  について使用率, 滞在時間等の計算

## 7. 演算量の比較

$Q_i$  におけるクラス  $j$  の客の平均待ち行列長  $L_{ij}(k)$   $i=1, 2, \dots, N$ ,  $j=1, \dots, L$  について計算するのに必要な演算量の比較を行なう.

SRadaw Node 法でも, (乗算回数) = (加算回数) なので乗算回数を基準とする. 等価流量法による計算法では, 前章の Step 1 から 4 までは全く同一であるが, Step 5~7 の部分が アルゴリズム F に

置き代る。従って、演算量比較は、アルゴリズムFと Step 5~7 について行えばよい。さらに、Step 6. の部分はアルゴリズムSの終了時点で自動的に得られるので、以下はアルゴリズムSと Step 7 の部分の乗算回数を調べ、収束までのループ回数を

$n$  とし、S5 の部分は Gauss-Jordan 法を使用するとすれば、アルゴリズムSの部分の乗算回数はおよそ、 $n\{R(L^2+L+1)+2L+L^3/2\}$  回となる。Step 7. の部分は、約  $NR(L+1)$  回となる。従って全体で次

$$(7.1) \quad (\text{乗算回数})_S = n\{R(L^2+L+1)+2L+L^3/2\} + NR(L+1).$$

ループ回数  $n$  が多ければ本稿の方法はアルゴリズムFに比べて非効率的、少なければ効率的となる。(7.1) と (4.2) と比較し、その臨界となる回数  $n^*$  および実測による回数  $n$  を表1に示す。

本例では、 $L=2$ ,  $N=5$ ,  $\varepsilon=0.001$  とした。

表1. に示されるように、網内容数  $K$  が増加しても収束ループ回数はあまり増加せず、本稿の方法は網の規模が大きくなるほど相対的に効率的な算法となっている。

Table 1. Critical iteration count  $n^*$  and observed iteration count  $n$  in the case of  $L=2$ ,  $N=5$ .

$K$	$n^*$	$n$
$K=(3,3)$	8.3 回	3 回
$K=(5,5)$	12.5	6
$K=(8,8)$	35.2	6

8. 数値計算例

8.1 セントラル・サーバ・モデル

図6. に示すモデルを例として、表2. に示すパラメータを設定し数値計算を行なう。

$Q_1, \dots, Q_N$  は  $N$  個のシングル・サーバで、先着順サービスとし、サービス時間は  $N$  個のパラメータ  $\mu_i = 0.5$  なる指数分布に従うものとする。

$m=1$ ,  $K=(3, 3)$  とし,  $\rho=(4.0, 1.0)$ ,  $\rho=(1.0, 4.0)$  とした時の計算結果を表3に示す。表3. においては,  $\alpha_{01}, \alpha_{02}$  及び  $E_{01}, E_{02}$  は各々  $Q_0$  におけるクラス1, 2の客の使用率と滞在時間,  $\alpha_0 = \alpha_{01} + \alpha_{02}$  と示す。  $E_1, E_2$  は各々クラス1, 2の客の網内一巡時間を示す。

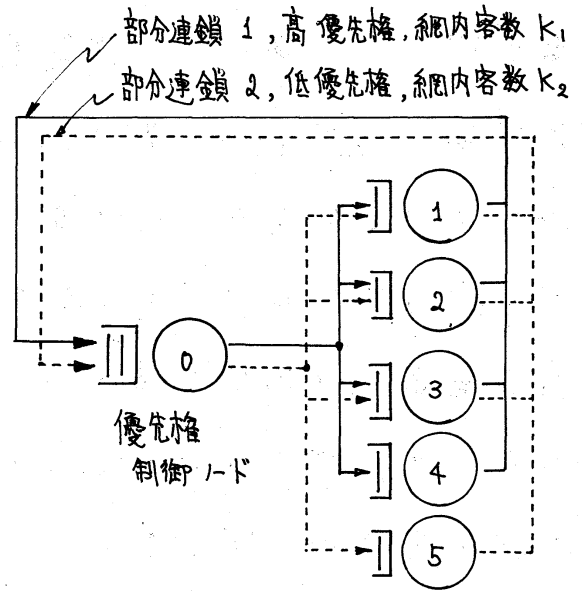


Figure 6. Central Server Model for numerical examples.

Table 2. Parameters for numerical examples.

Node No. (i)	$\theta_{ij}$		$\rho_{ij} = \theta_{ij} / \mu_i$	
	1	2	1	2
0	1	1	—	—
1	2	3	1.0	1.5
2	3	2	1.5	1.0
3	4	4	2.0	2.0
4	3	0	1.5	0
5	0	3	0	1.5

$\mu_i = 0.5 \quad (i=1, 2, \dots, 5)$



Table 3. Numerical examples with  $K=(3, 3)$ 

$\rho=(\rho_1, \rho_2)$	使用率 (%)			滞在時間				
	$Q_0$			$Q_0$		全体		
	$Q_{01}$	$Q_{02}$	$Q_0$	$E_{01}$	$E_{02}$	$E_1$	$E_2$	
(4.0, 1.0)	優	70	12	82	7.7	15.4	17.1	25.1
	劣	58	22	80	9.5	2.9	20.6	13.5
(1.0, 4.0)	優	25	55	80	1.2	10.4	12.0	21.8
	劣	22	58	80	2.9	9.5	20.6	13.5

8.2 サーバ数  $M$  による滞在時間の変化

5.5 節に示した Conjecture を確認するための数値実験例を  
表4. に示す。モデル及びパラメータは前節に準ずる。本例では、  
 $K=(3, 3)$ ,  $\rho=(2.5, 2.5)$  とした。 $Q_0$  のサーバ数  $m$  について、 $m=1$   
及び  $m=3$  の場合を想定し、 $Q_s$  のサーバ数を  $M=1, 3, \infty$  と変化さ  
せた場合の比較を行なった。

$E_{s1}, E_{s2}$  は各々  $Q_s$  にお  
けるクラス 1, 2 の滞在時  
間を示すものとし、 $E_{01},$   
 $E_{02}$  は  $Q_0$  における滞在  
時間とする。 $Q_0$  にお  
ける滞在時間に最も近  
い滞在時間を示す  $M$  を

Table 4. Sojourn time at node  $Q_s$ .

$M$	$m=1$		$m=3$	
	$E_{s1}$	$E_{s2}$	$E_{s1}$	$E_{s2}$
$M=1$	4.0	13.9	2.27	2.42
$Q_s: M=3$	4.3	14.4	2.49	2.65
$M=\infty$	4.4	14.7	2.53	2.69
$Q_0: E_{0j}$	3.9	13.4	2.50	2.64

も  $Q_s$  が最適ロード  $Q_s^*$  となる。表 4. が示すように、本例では先の Conjecture の成立は確認される。

## 9. まとめ

閉鎖型待ち行列網において、1 個のロードが優先権制御を行なうような待ち行列網の近似解法について述べた。

本稿では、積形式解を破るロードとして優先権制御ロードを想定して論を進めたが、本稿の方法は等価流量法を用いる近似解法一般に適用する事が可能である。

本稿ではいくつかの数值計算例を示したが、なおシミュレーション結果との比較による精度の検証は今後の課題として残されている。さらに、積形式解を破るロードが複数個存在する網、開放型部分連鎖の加わり、た混合型待ち行列網へとその応用範囲を拡張することも今後の課題として残されている。

## 【参考文献】

- 1) Chandy, K.M., Herzog, U., Woo, L, Parametric Analysis of Queueing Networks, IBM J. Res. Dev., vol. 19, no. 1, 1975.
- 2) Marie, R.A., An Approximate Analytical Method for General Queueing Networks, Trans. IEE, vol. SE-5, no. 5, 1979.
- 3) Neuse, D and Chandy, K.M., HAM: The heuristic aggregation method for solving General Closed Queueing Network Models of Computer Systems, Performance Evaluation Review, 11, 1982.
- 4) Sevcik, K.C., Priority Scheduling Discipline in Queueing Models of Computer Systems, Proc. IFIP, 1977.
- 5) Chow, W.M. and Yu, P.S., An Approximation Technique for Central Server Queueing Models with a Priority Dispatching Rule, Performance Evaluation 3 (1983), 55-62.
- 6) BusBett, F., Chandy, K.M., Murty, R.R., and Palacios, F.G., Open Closed and Mixed Networks of Queues with Different Classes of Customers, JACM, vol. 22, no. 2, 1975.
- 7) Reiser, M. and Kobayashi, H, Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms, IBM J. Res. Dev., vol. 19, no. 3, 1975